

Latest Version: 13.0

Question: 1

What are two characteristics of RPC API calls? (Choose two.)

- A. They can be used only on network devices.
- B. They use only UDP for communications.
- C. Parameters can be passed to the calls.
- D. They must use SSL/TLS.
- E. They call a single function or service.

Answer: CE

Explanation:

Reference: <https://pubs.opengroup.org/onlinepubs/9629399/chap6.htm>

Question: 2

Which two actions do Python virtual environments allow users to perform? (Choose two.)

- A. Simplify the CI/CD pipeline when checking a project into a version control system, such as Git.
- B. Efficiently port code between different languages, such as JavaScript and Python.
- C. Run and simulate other operating systems within a development environment.
- D. Quickly create any Python environment for testing and debugging purposes.
- E. Quickly create an isolated Python environment with module dependencies.

Answer: DE

Explanation:

Reference: <https://realpython.com/python-virtual-environments-a-primer/>

Question: 3

What are two benefits of leveraging Ansible for automation of Cisco IOS XE Software? (Choose two.)

- A. Ansible playbooks are packaged and installed on IOS XE devices for automatic execution when an IOS device reboots.
- B. All IOS XE operating systems include Ansible playbooks for basic system administration tasks.
- C. It is a device-independent method for automation and can be used with any type of device or operating system.

- D. Ansible playbooks can be written from the IOS XE EXEC command line to configure the device itself.
- E. It does not require any modules of software except SSH to be loaded on the network device.

Answer: CE

Explanation:

Reference: <https://developer.cisco.com/learning/modules/intro-ansible-iosxe/ansible-overview/step/4>

Question: 4

Refer to the exhibit.

```
return_val=
{
  "alertId": "643451796765672516",
  "alertType": "appliances went down",
  "deviceMac": "e0:55:3d:6c:c1:7a",
  "deviceName": "MX65 c1:7a",
  "deviceSerial": "Q2QN-58EA-XXXX",
  "deviceUrl": "https://n143.meraki.com/Branch-1/n/.../manage/nodes/new_wired_status",
  "networkId": "L_1234567890",
  "networkName": "Branch 1",
  "networkUrl": "https://n143.meraki.com/Branch-1/n/.../manage/nodes/wired_status",
  "occuredAt": "2018-11-10T18:45:20.000000Z",
  "organizationId": "1234567",
  "organizationName": "Meraki Demo",
  "organizationUrl": "https://n143.meraki.com/o/.../manage/organization/overview",
  "sentAt": "2018-11-10T18:50:30.479982Z",
  "SharedSecret": "asdf1234",
  "version": "0.1"
}
```

The task is to create a Python script to display an alert message when a Meraki MX Security Appliance goes down. The exhibit shows sample data that is received. Which Python snippet displays the device name and the time at which the switch went down?

- A.

```
with return_val:
    print("The Switch: "+deviceName+ ",
    went down at: "+occurredAt)
```
- B.

```
print("The Switch: "+return_val.deviceName+ ", \
went down at: "+return_val.occurredAt)
```
- C.

```
print("The Switch: "+return_val['deviceName']+ ", \
went down at: "+return_val['occurredAt'])
```
- D.

```
with items as return_val:
    print("The Switch: "+items.deviceName+ ",
    went down at: "+items.occurredAt)
```

- A. Option A
- B. Option B
- C. Option C
- D. Option D

Answer: C

Question: 5

Refer to the exhibit.

```
{
  "alertData": {
    "countNode": 1,
    "bssids": [
      "aa:bb:cc:dd:ee:ff",
      "11:22:33:44:55:66"
    ],
    "minFirstSeen": 1548512334,
    "maxLastSeen": 1548512802,
    "countIsContained": 0,
    "reason": "Seen on LAN",
    "wiredMac": "aa:bb:cc:dd:ee:f0"
  },
  "alertId": "629378047939282802",
  "alertType": "Air Marshal -Rogue AP detected",
  "occuredAt": "2019-01-26T14:18:54.000000Z",
  "organizationId": "123456",
  "organizationName": "Organization",
  "organizationUrl": "https://nl.meraki.com/o/.../manage/organization/overview",
  "networkId": "L_123456789012345678",
  "networkName": "Network",
  "networkUrl": "https://nl.meraki.com/.../manage/nodes/list",
  "version": "0.1"
  "SharedSecret": "supersecret",
  "sentAt": "2019-01-26T14:35:20.442869Z",
}
```

The goal is to write a Python script to automatically send a message to an external messaging application

when a rogue AP is detected on the network. The message should include the broadcast SSID that is in the alert. A function called "send_to_application" is created, and this is the declaration:

```
send_to_application(message)
```

The exhibit also shows the data that is received by the application and stored in the variable return_val.

Which Python code completes the task?

- A. `bssids =return_val["bssids"]`
`for number in range(return_val["alertData"]["countNode"]):`
`send_to_application ("ALERT: detected a bssid on the`
`network: "+ return_val["alertData"][bssids][number])`
- B. `bssids =return_val["bssids"]`
`for value in bssids:`
`send_to_application ("ALERT: detected a bssid on the`
`network: "+value)`
- C. `count = return_val["alertData"]["countNode"]`
`bssids =return_val["alertData"][count]["bssids"]`
`for value in bssids:`
`send_to_application ("ALERT: detected a bssid on the`
`network: "+value)`
- D. `bssids =return_val["alertData"]["bssids"]`
`for value in bssids:`
`send_to_application ("ALERT: detected a bssid on the`
`network: "+value)`

- A. Option A
B. Option B
C. Option C
D. Option D

Answer: D

Explanation:

For number in range value is required for the application to send the alert. Bssids are also included.